

## **REMARKS/ARGUMENTS**

Applicants acknowledge receipt of Examiner voice message, on June 19, 2008, that the current action in the application is a Non-Final action. Applicants respond accordingly.

Claims 1, 3, 5, 8, 10, 21 and 22 are pending in the present application. Claims 1, 5, 10, 21, and 22 are amended. Claims 2, 4, 6-7, and 11-20 are canceled. Support for the claim amendments can be found in the claims as originally filed, as the claim amendments incorporate features from previously presented dependent claims. No new matter is added. Reconsideration of the claims is respectfully requested.

Applicants have amended some claims and canceled others. Applicants do not concede that the subject matter encompassed by the earlier presented claims is not patentable over the art cited by the Examiner. Applicants canceled claims in this response solely to facilitate expeditious prosecution of this application. Applicants traverse all rejections and respectfully reserve the right to pursue the earlier-presented claims including subject matter encompassed by the claims prior to this amendment, and additional claims, in one or more continuing applications.

### **I. 35 U.S.C. § 103, Obviousness**

The Examiner has rejected claims 1, 3-6, 8-10, 21 and 22 under 35 U.S.C. § 103 as being unpatentable over *Carlson*, U.S. Patent Number 6,697,849, published February 24, 2004, hereinafter "*Carlson*", in view of *Johnson*, "The Application Response Measurement API, Version 2" Tivoli Systems, December 1997, cited in a related case by Applicant, hereinafter "*Johnson*". This rejection is respectfully traversed.

In rejecting the claims, the Examiner states:

Referring to claim 1, Carlson discloses a method of distributing traffic to application instances (i.e. applications 202-208 running on application server 200) on one or more computing devices (i.e. servers 308A-C), comprising:

- obtaining application instance specific operational information (i.e. server load criteria and application component performance criteria) identifying operational characteristics (i.e. elements shown in Figures 8 and 9) of an application instance on a computing device on the one or more computing devices (e.g. abstract; col. 12, lines 40-67);

- generating a load balancing weight to be associated with an application instance based on the application instance specific operational information (i.e. random number is generated in a weighted manner according to the "best" server at that particular time) (col. 16, lines 13-47); and

- distributing traffic based on the generated load balancing weight (i.e. "gracefully" distribute requests among the application servers) (col. 16, lines 35-47).

Carlson does not explicitly disclose that the instance specific operational information includes an application instance topology and the topology is obtained from the instance using an agent application residing on the computing device, and

wherein the agent application identifies the application instance topology by sending a correlation in a request to an agent application associated with a second application instance, wherein application instance information is provided by the agent application associated with the second application. In analogous art, Johnson discloses another transaction processing system which discloses obtaining application topology information using an agent application residing on the computing device, and wherein the agent application identifies the application instance topology by sending a correlation in a request to an agent application associated with a second application instance, wherein application instance information is provided by the agent application associated with the second application (i.e. at the start of a transaction, the application can provide a correlator for a parent transaction to an ARM agent program, the correlation application can then collect all the information regarding these transactions and from which applications they came from, and then get the total picture regarding where transactions come from and what applications they interact with (pp. 7-9; "Using ARM to Correlate Transactions and Subtransactions"). It would have been obvious to one of ordinary skill in the art to combine the teaching of Johnson with Carlson in order to utilize the Application resource monitoring system of Johnson which monitors application resources (i.e. thresholds can be defined and monitored and a notification can be sent to an automation routine when congestion is detected; Johnson, p. 9) with the performance criteria used by Carlson, thereby increasing the ability to customize load balancing weights according to the user's liking, and to minimize congestion within the application.

Office Action dated 03/27/08, pp. 2-4.

The Examiner bears the burden of establishing a *prima facie* case of obviousness based on prior art when rejecting claims under 35 U.S.C. § 103. *In re Frütz*, 972 F.2d 1260, 23 U.S.P.Q.2d 1780 (Fed. Cir. 1992). The prior art reference (or references when combined) must teach or suggest all the claim limitations. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974). In determining obviousness, the scope and content of the prior art are... determined; differences between the prior art and the claims at issue are... ascertained; and the level of ordinary skill in the pertinent art resolved. Against this background the obviousness or non-obviousness of the subject matter is determined. *Graham v. John Deere Co.*, 383 U.S. 1 (1966). "Often, it will be necessary for a court to look to interrelated teachings of multiple patents; the effects of demands known to the design community or present in the marketplace; and the background knowledge possessed by a person having ordinary skill in the art, all in order to determine whether there was an apparent reason to combine the known elements in the fashion claimed by the patent at issue." *KSR Int'l. Co. v. Teleflex, Inc.*, No. 04-1350 (U.S. Apr. 30, 2007). "*Rejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness.*" *Id.* (citing *In re Kahn*, 441 F.3d 977, 988 (CA Fed. 2006))."

**I.A. The Proposed Combination of References Does not Teach or Suggest All the Features the Claims**

The proposed combination of references, when considered as a whole, does not teach or suggest each and every feature of the claims. By the present Amendment, Applicants have canceled claims 2, 4, 6-7, and 11-20. Therefore, the rejection with respect to these claims is moot. Applicants have amended claim 1 to incorporate the features of claims 2, 4, 6-7, and 11-20. Amended claim 1 is as follows:

1. A method, in a data processing system, of distributing traffic to application instances on one or more computing devices, comprising:
  - obtaining application instance specific operational information identifying operational characteristics of an application instance on a computing device of the one or more computing devices, wherein the application instance specific operational information includes at least one of an application instance topology, an importance of transactions currently being processed by the application instance, an amount of time the application instance has been blocked waiting for resources, and an amount of resources consumed by the application instance;
  - comparing the application instance specific operational information to one or more other application instance specific operational information for one or more other application instances based on the application instance specific operational information obtained;
  - generating a load balancing weight based on a relationship between the application instance specific operational information and the one or more other application instance specific operational information;
  - attributing weight points to the application instance and the one or more other application instances based on a relative difference between the application instance specific operational information and the one or more other application instance specific operational information;
  - distributing the traffic to the application instance based on the load balancing weight;
  - wherein obtaining application instance specific operational information includes retrieving the application instance specific operational information from the application instance using an agent application residing on the computing device, and wherein the agent application identifies the application instance topology by sending a correlation in a request to an agent application associated with a second application instance, wherein application instance information is provided by the agent application associated with the second application; and
  - wherein the method is implemented in a weight management system that is separate from the computing devices and from a load balancing device.

Neither *Carlson* nor *Johnson* teaches or suggest the features “comparing the application instance specific operational information to one or more other application instance specific operational information for one or more other application instances based on the application instance specific operational information obtained” and “generating a load balancing weight based on a relationship between the application instance specific operational information and the one or more other application instance specific operational information.”

*Carlson* teaches a method for caching JSP component request responses. The component request responses are cached and managed by a JSP engine. When a client request referencing a JSP component is received, the JSP engine first checks the response cache to determine whether a valid response is available. If a valid response is found, the response is streamed to the client; where a match is not found, the JSP component is executed.

*Johnson* describes a method for managing and tracking business transactions by embedding ARM API program calls to retrieve critical information about business transactions in terms of the business operations. *Johnson* teaches that implementation of this method requires modification of an application to included the ARM API programs calls. However, neither *Carlson* nor *Johnson* teaches or suggests that application instance specific operational information is retrieved from an application instance such that the information that is retrieved is used in a comparison or in generating an appropriate load balancing solution. Accordingly, because neither *Carlson* nor *Johnson* teach or suggest all the features of claim 1, the proposed combination of *Carlson* nor *Johnson* when considered as a whole does not teach or suggest all the features of claim 1. Therefore, no *prima facie* obviousness rejection can be stated against claim 1.

#### **I.B. The Proposed Combination of References Changes the Principle Operation of the Primary Reference**

Additionally, the Examiner failed to state a *prima facie* obviousness rejection because the proposed combination changes the principle of operation of the primary reference. Obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness. See *KSR Int'l. Co. v. Teleflex, Inc.*, No. 04-1350 (U.S. Apr. 30, 2007)(citing *In re Kahn*, 441 F.3d 977, 988 (CA Fed. 2006)). In combining references to show the claimed feature, the proposed modification cannot change the principle of operation of a reference. See *In re Ratti*, 270 F.2d 810, 123 (CCPA 1959) and MPEP 2143.01. If the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims *prima facie* obvious. *Id.*

As previously discussed *Carlson* teaches a method for caching JSP component request responses. *Johnson* states:

There are three steps to monitoring application performance with the ARM API.

1. The first step is to define the key business transactions. This is the most important step. Each application developer needs to define who needs what kind of data, and what the data will be used for. It is common and useful for this process to be a joint collaboration between the users and developers of an application, and system and network administrators.

2. The second step is to ***modify the program to include calls to the ARM API***. The stub libraries and logging agent in the developers toolkit can be used for initial testing. Because the API calls are simple, this step is neither difficult nor time-consuming. The key is to decide where to put the probes in the first place, by doing a good job defining the key business transactions.
3. The third step is to replace the stub libraries from the developers toolkit with an ARM-compliant agent and associated management applications. The distributed applications will now be monitored in ways that could previously only be hoped for.

*Johnson*, page 3 column 2, How to Use the API section

The functionality taught in the above cited section could not be implemented in *Carlson*, without requiring that each JSP component be modified to include calls to the ARM API. Implementation of the feature taught in *Johnson* would result in delayed processing of the JSP component request. The delayed processing would result in an increase in the overall overhead of the system. A person of ordinary skill in the art would understand that such an implementation would modify the principle operation of *Carlson*, which is to cache JSP component request responses so they can be quickly retrieved for future processing. As shown above, *In re Ratti* provides that changing the principle operation of a device renders a claim non-obvious in view of the proposed combination. Therefore, the Examiner failed to state a *prima facie* obviousness rejection against claims in the current invention.

## **II. Dependent Claims**

The remaining claims depend from claim 1. Consequently, because the remaining claims depend from claim 1, the combination of references, when considered as a whole, does not teach or suggest all of the features of the dependant claims. Thus under the standards of *In re Royka*, the Examiner fails to state a *prima facie* obviousness rejection of claims 3, 5, 8, 10, 21, and 22. Therefore the rejection of 3, 5, 8, 21, and 22 under 35 U.S.C. § 103 has been overcome.

### III. Conclusion

It is respectfully urged that the subject application is patentable over the cited references and is now in condition for allowance.

The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: June 20, 2008

Respectfully submitted,

/Gerald H. Glanzman/

Gerald H. Glanzman  
Reg. No. 25,035  
Yee & Associates, P.C.  
P.O. Box 802333  
Dallas, TX 75380  
(972) 385-8777  
Attorney for Applicants